Vorbereitung auf die 2.Teilprüfung

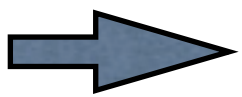
Stoff der 2. Teilprüfung

- Algorithmus entwerfen, implementieren
- Collections (ArrayList, Stack, Maps)
- Containertypen: String, Arrays
- Interfaces
- Rekursion

Beispiele

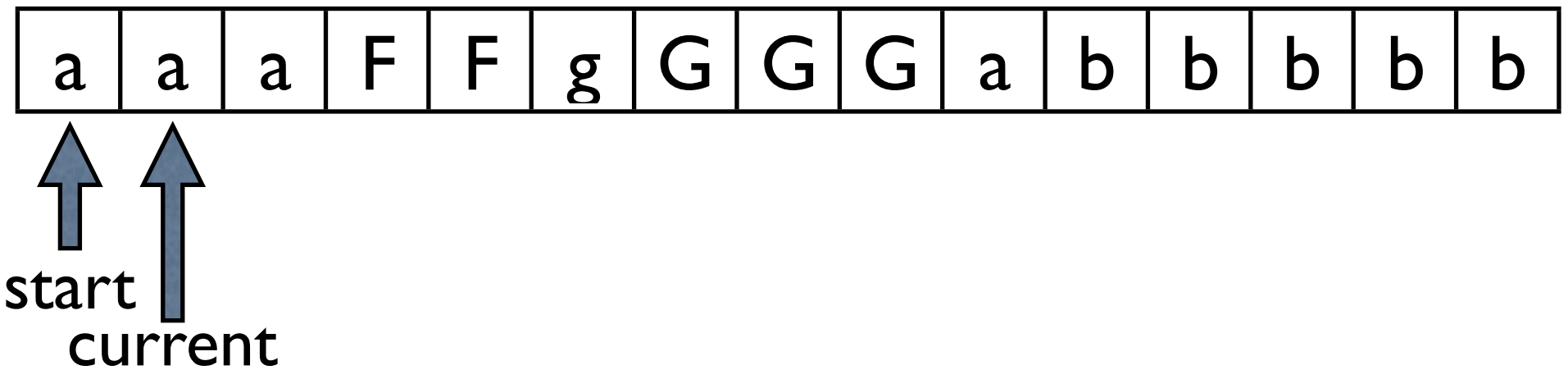
- Aufgabe 1: Algorithmus, String
- Aufgabe 2: Klassen und Interfaces schreiben
- Aufgabe 1/2: Variante mit Rekursion

Aufgabe 1

- Umgang mit Strings
- Lauflängenkodierung (Runlength Encoding)
- „aaaFFgGGGabbbbbb“  „3a2Fg3Ga5b“

Lauflängenkodierung

internal

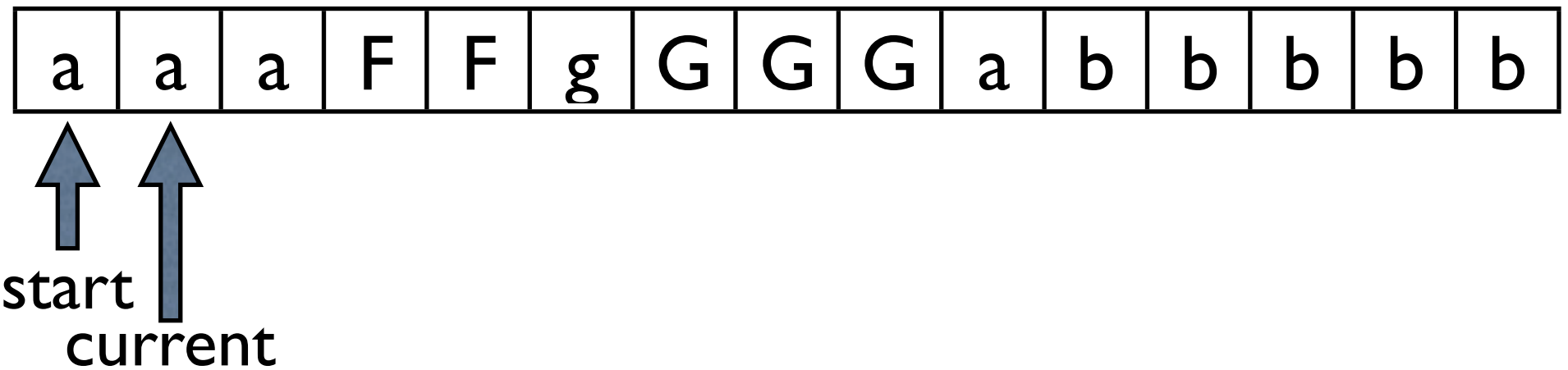


repeat = 0

compressed= null

Lauflängenkodierung

internal

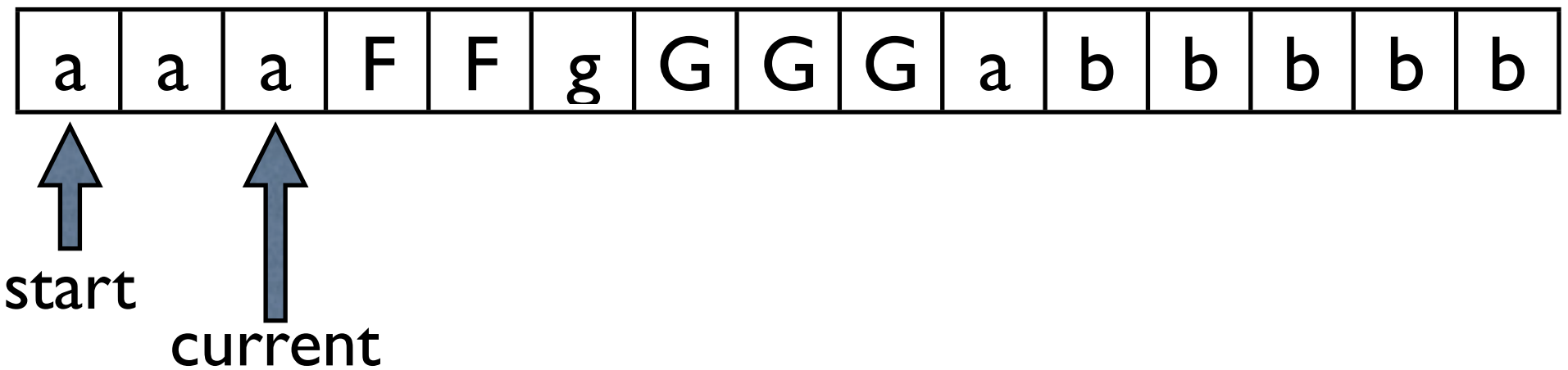


repeat = 1

compressed= null

Lauflängenkodierung

internal

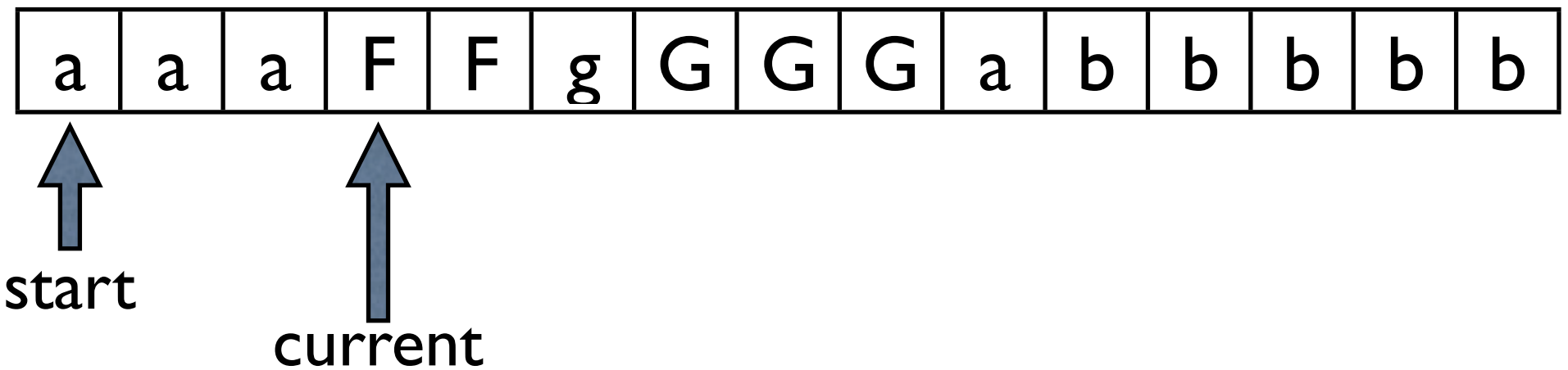


repeat = 2

compressed= null

Lauflängenkodierung

internal

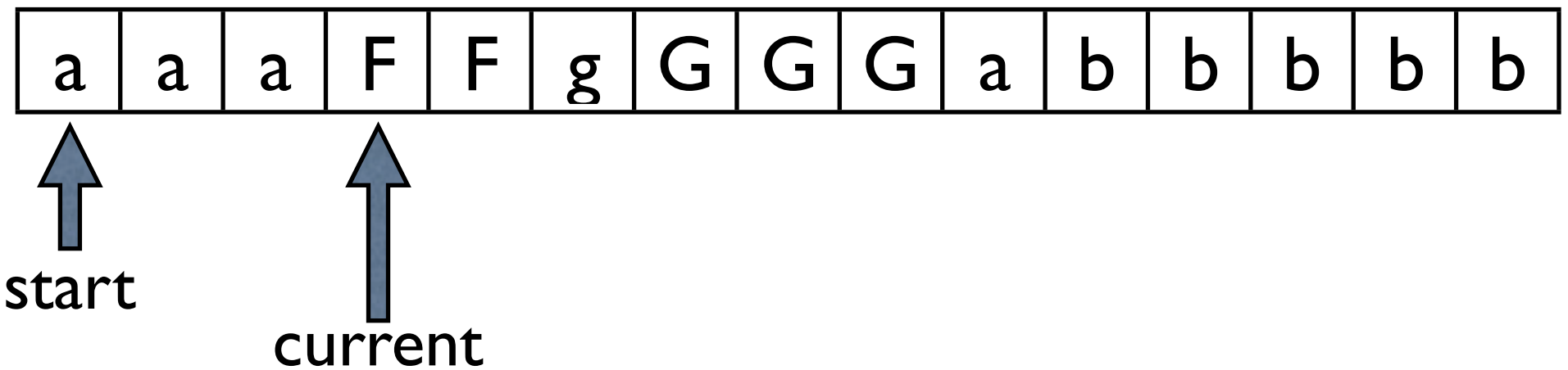


repeat = 2

compressed= 3 a

Lauflängenkodierung

internal

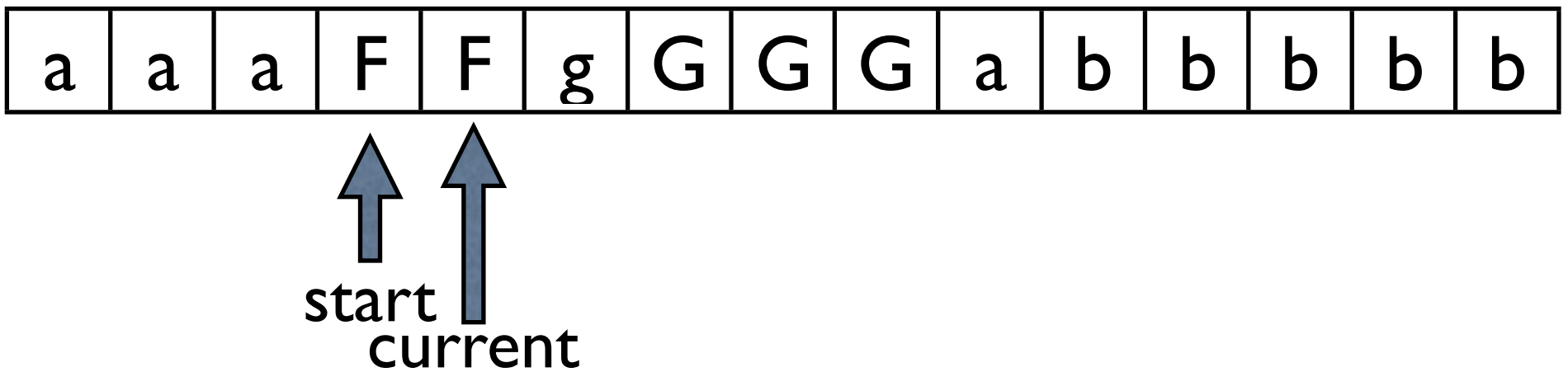


repeat = 0

compressed= | 3 | a |

Lauflängenkodierung

internal



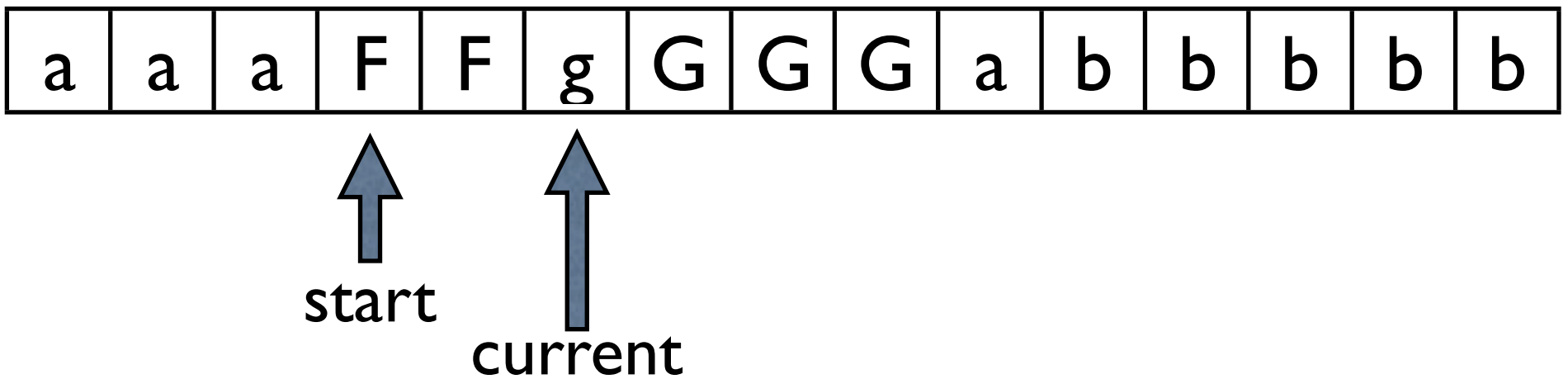
repeat =

compressed=

3	a

Lauflängenkodierung

internal



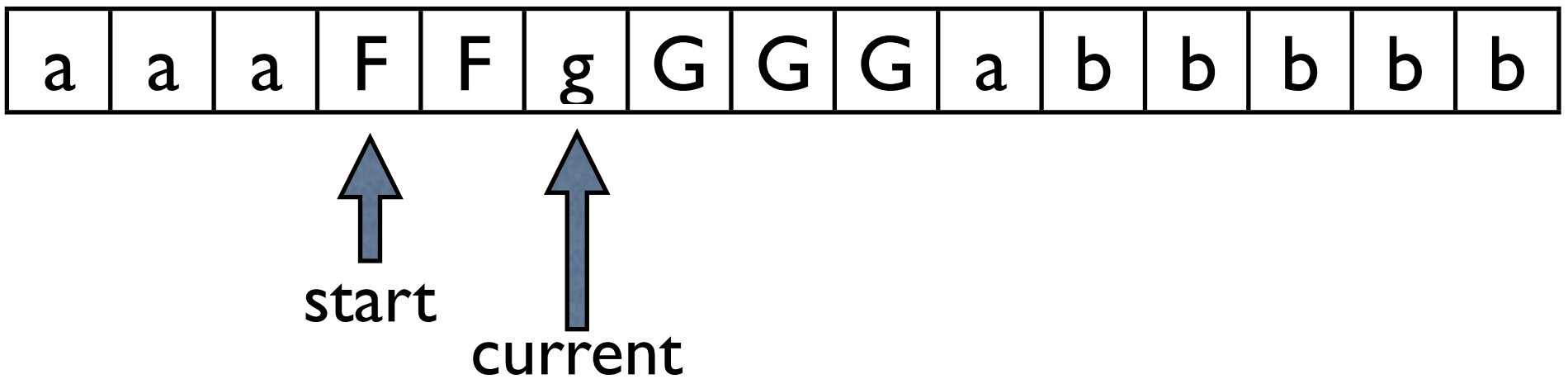
repeat =

compressed=

3	a	2	F

Lauflängenkodierung

internal



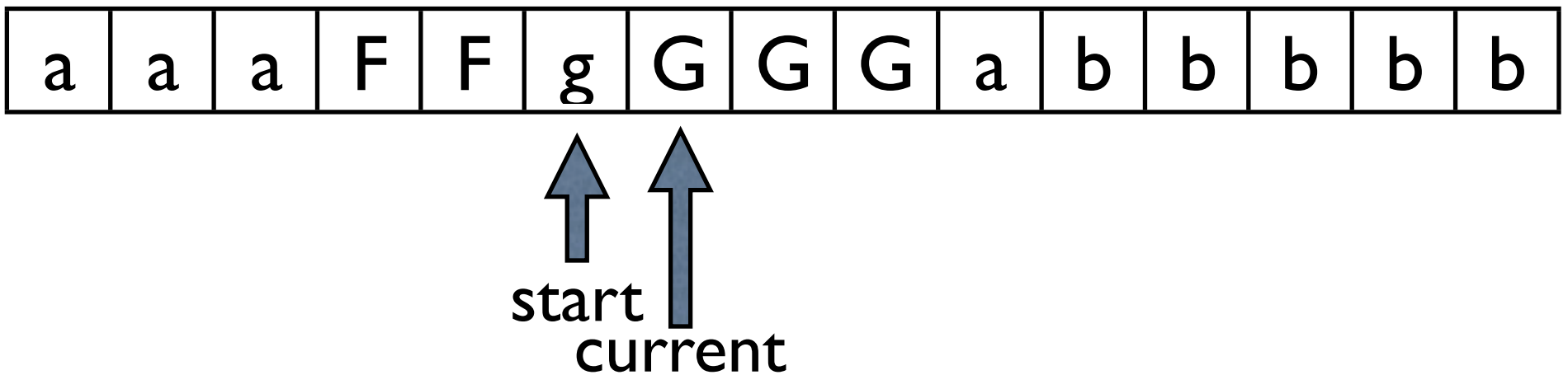
repeat = 0

compressed=

3	a	2	F

Lauflängenkodierung

internal



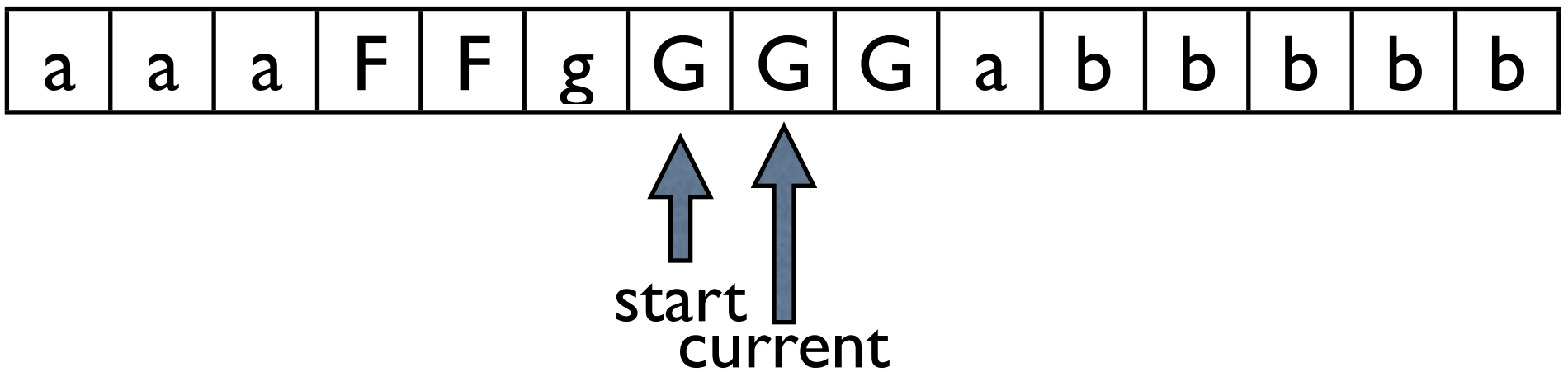
repeat = 0

compressed=

3	a	2	F	g

Lauflängenkodierung

internal



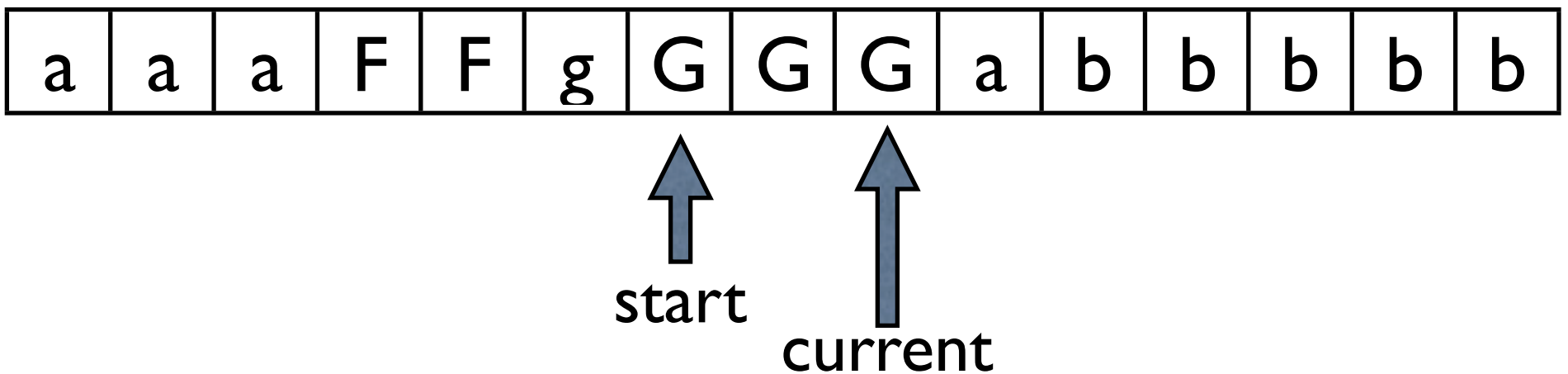
repeat =

compressed=

3	a	2	F	g

Lauflängenkodierung

internal



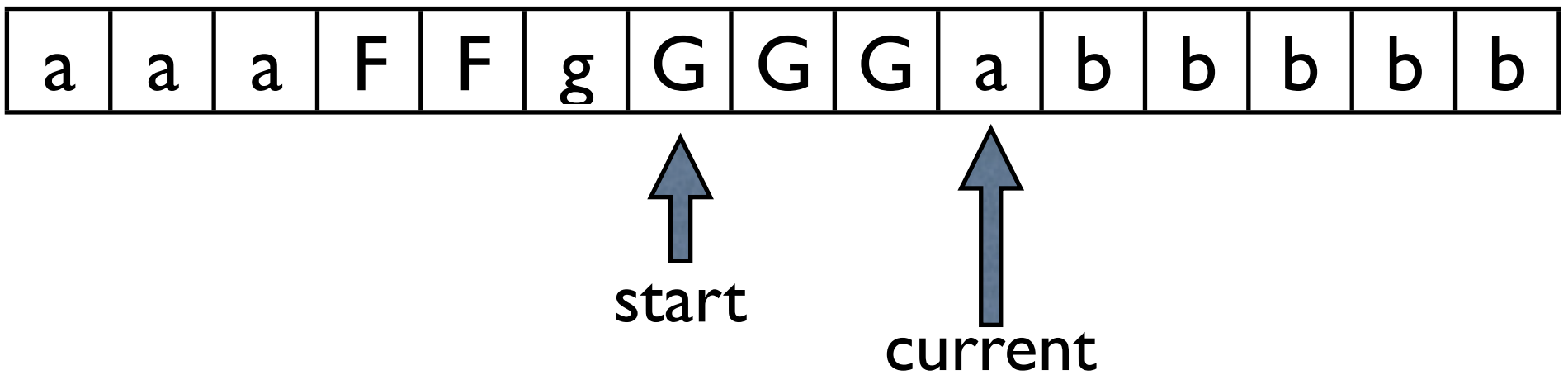
repeat = 2

compressed=

3	a	2	F	g

Lauflängenkodierung

internal



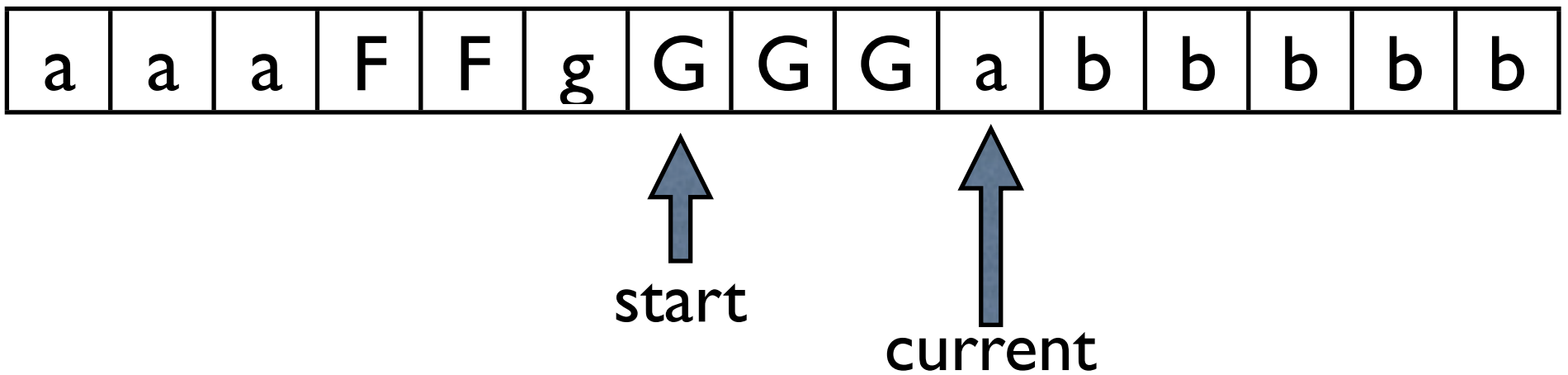
repeat = 2

compressed=

3	a	2	F	g

Lauflängenkodierung

internal



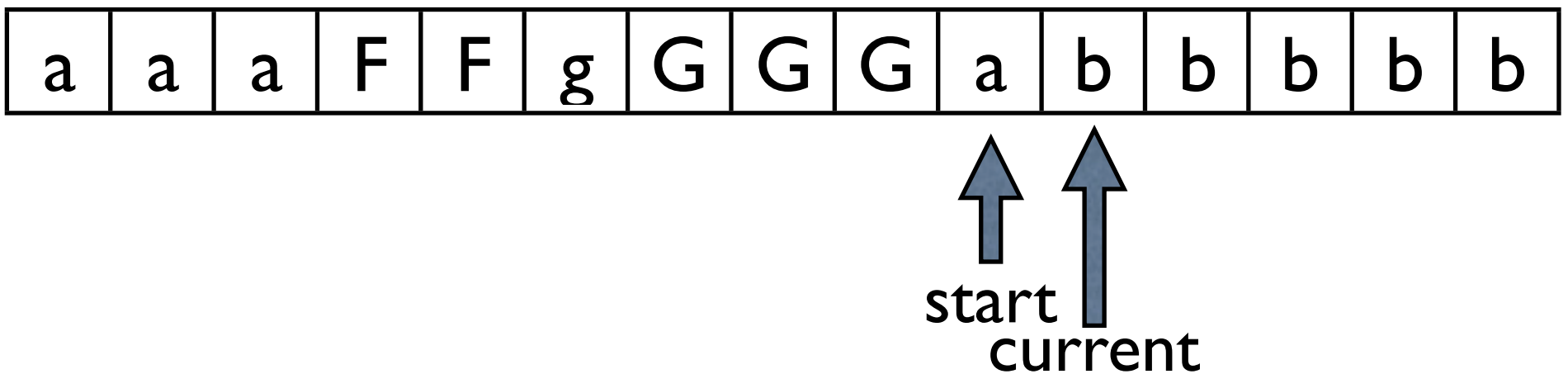
repeat = 0

compressed=

3	a	2	F	g

Lauflängenkodierung

internal



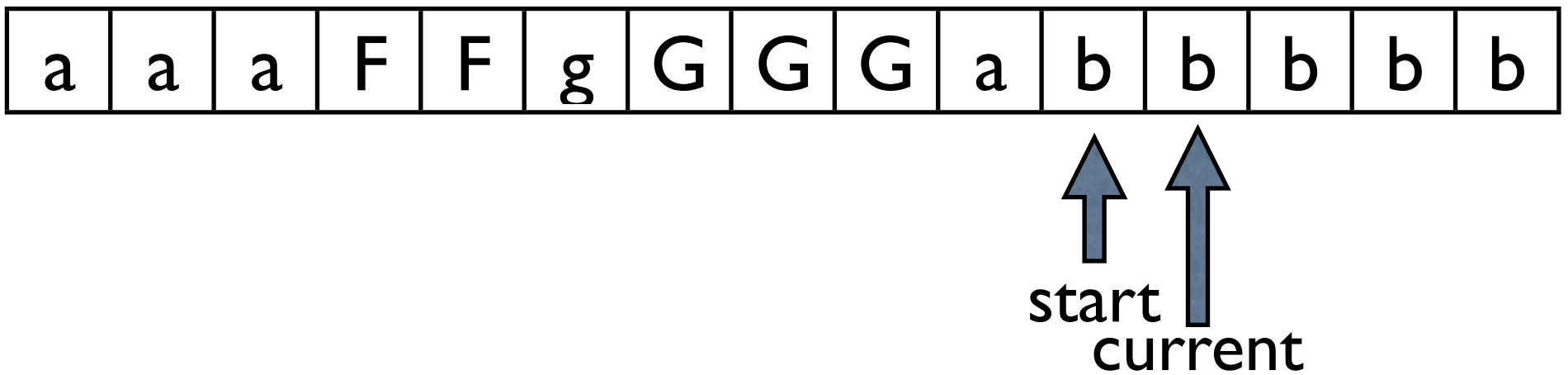
repeat = 0

compressed=

3	a	2	F	g	3	G	a

Lauflängenkodierung

internal



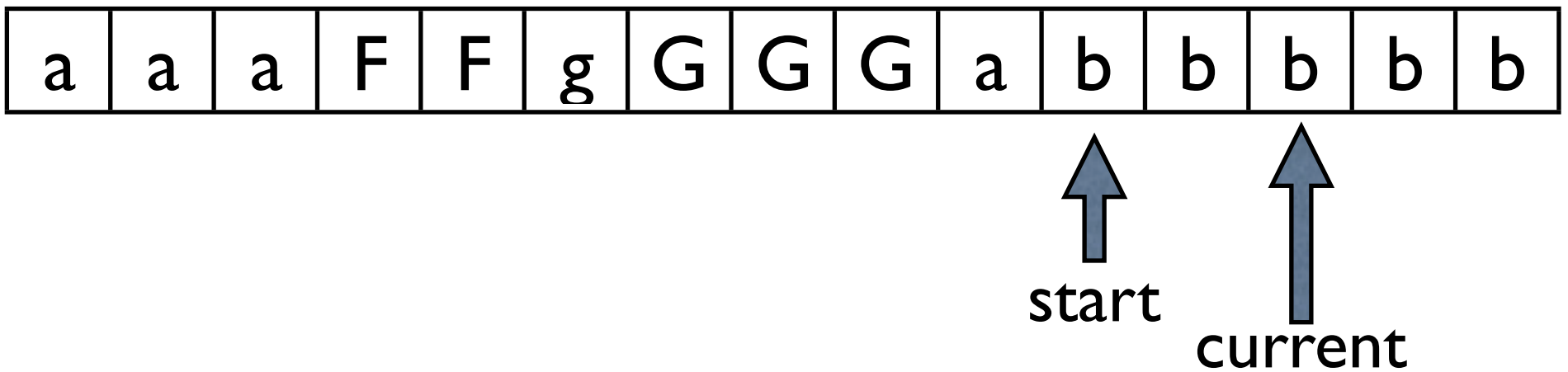
repeat =

compressed=

3	a	2	F	g	3	G	a

Lauflängenkodierung

internal



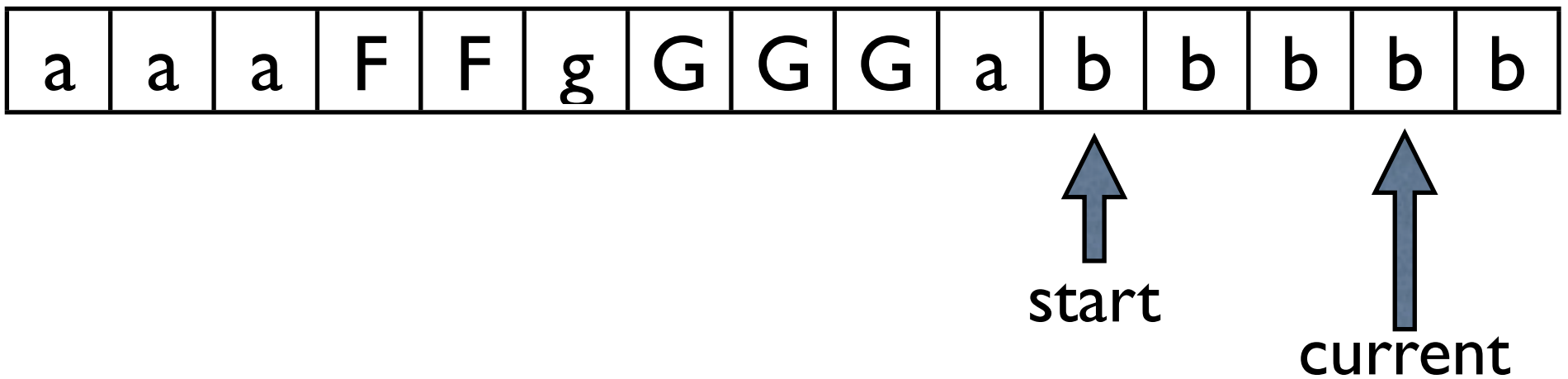
repeat = 2

compressed=

3	a	2	F	g	3	G	a

Lauflängenkodierung

internal



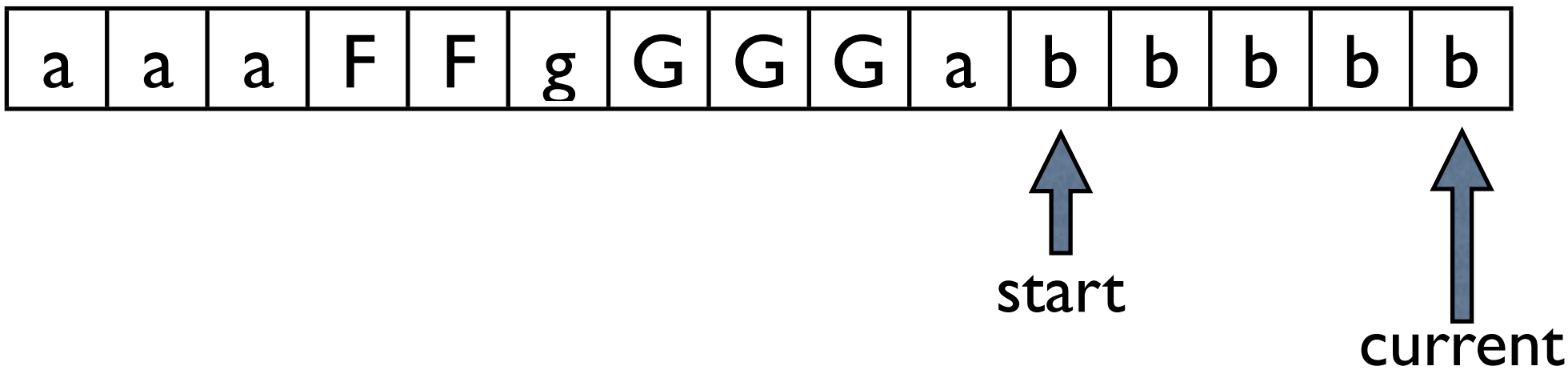
repeat = 3

compressed=

3	a	2	F	g	3	G	a

Lauflängenkodierung

internal



repeat = 4

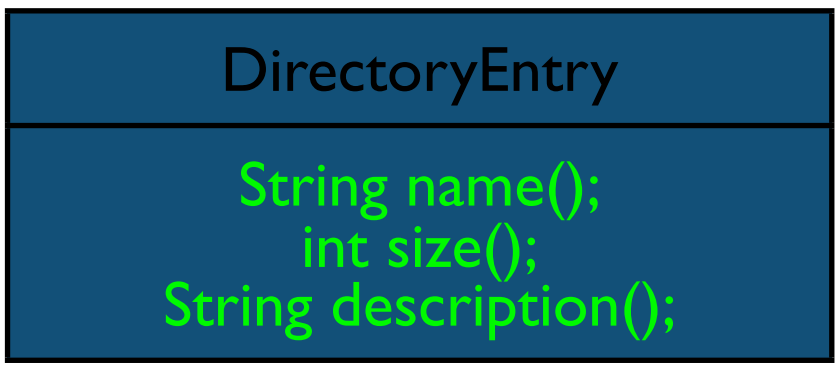
compressed=

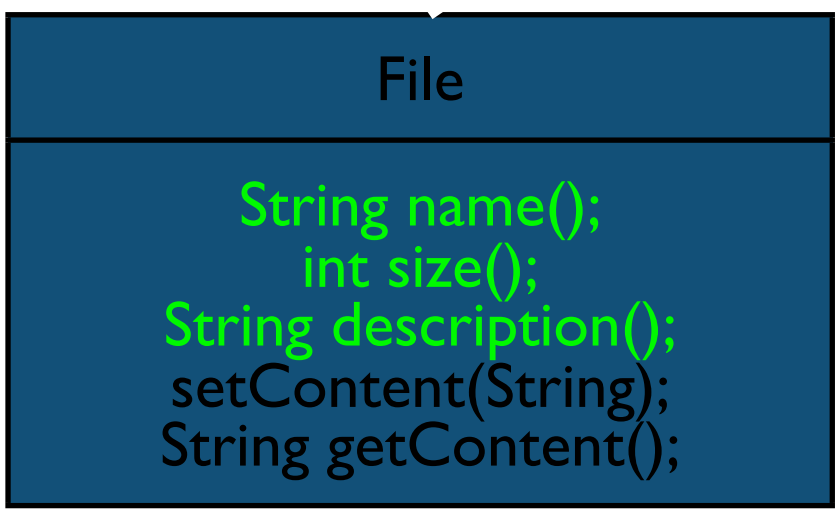
3	a	2	F	g	3	G	a	5	b

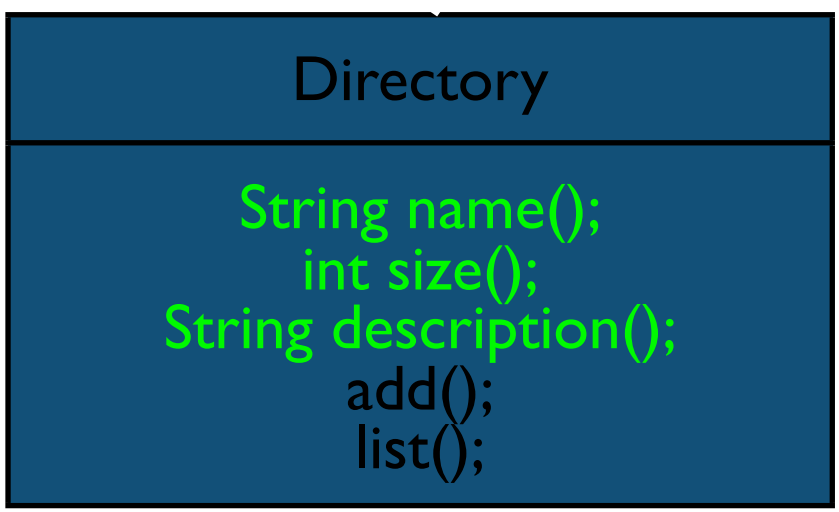
Beispiel Dateisystem

- Interface: `DirectoryEntry`
- Klassen: `File, Directory`

Beispiel: Dateisystem

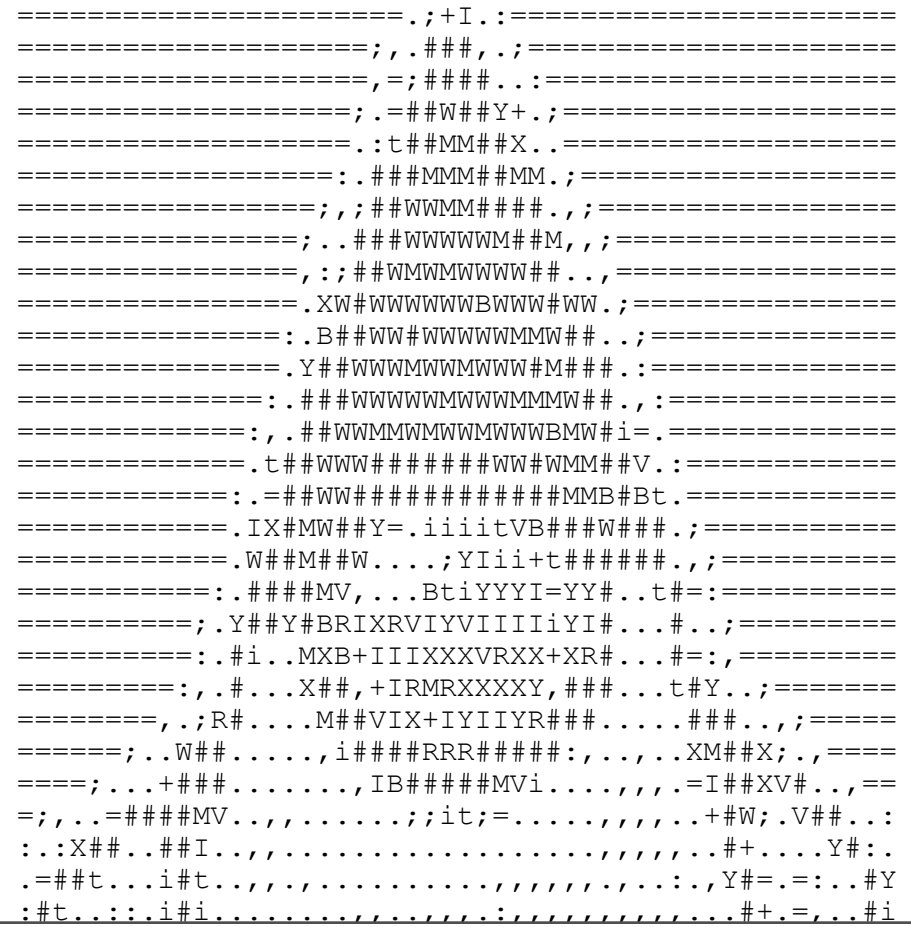






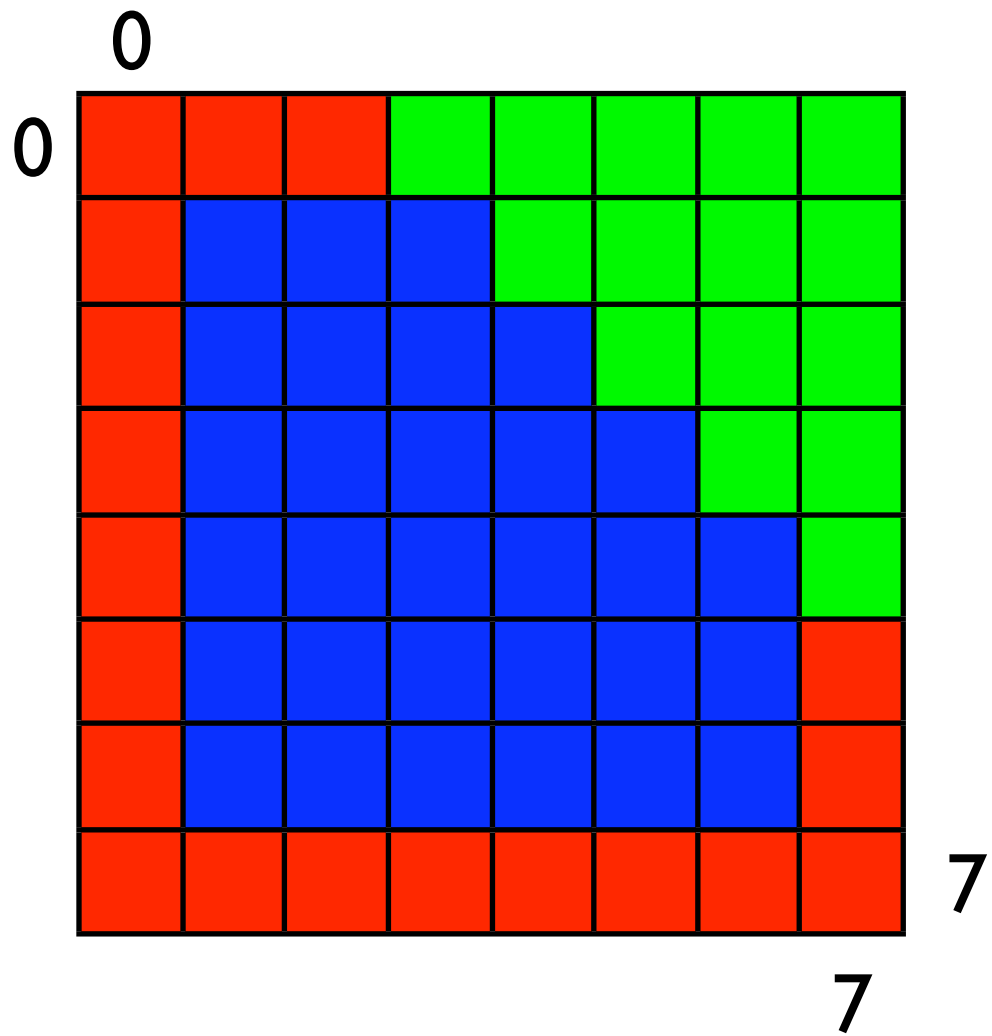
Beispiel: „Region Growing“

- Implementieren eines Region Growing Algorithmus
- Inhalte:
  - Klassen
  - Umgang mit Arrays
  - Rekursion



„Region Growing“

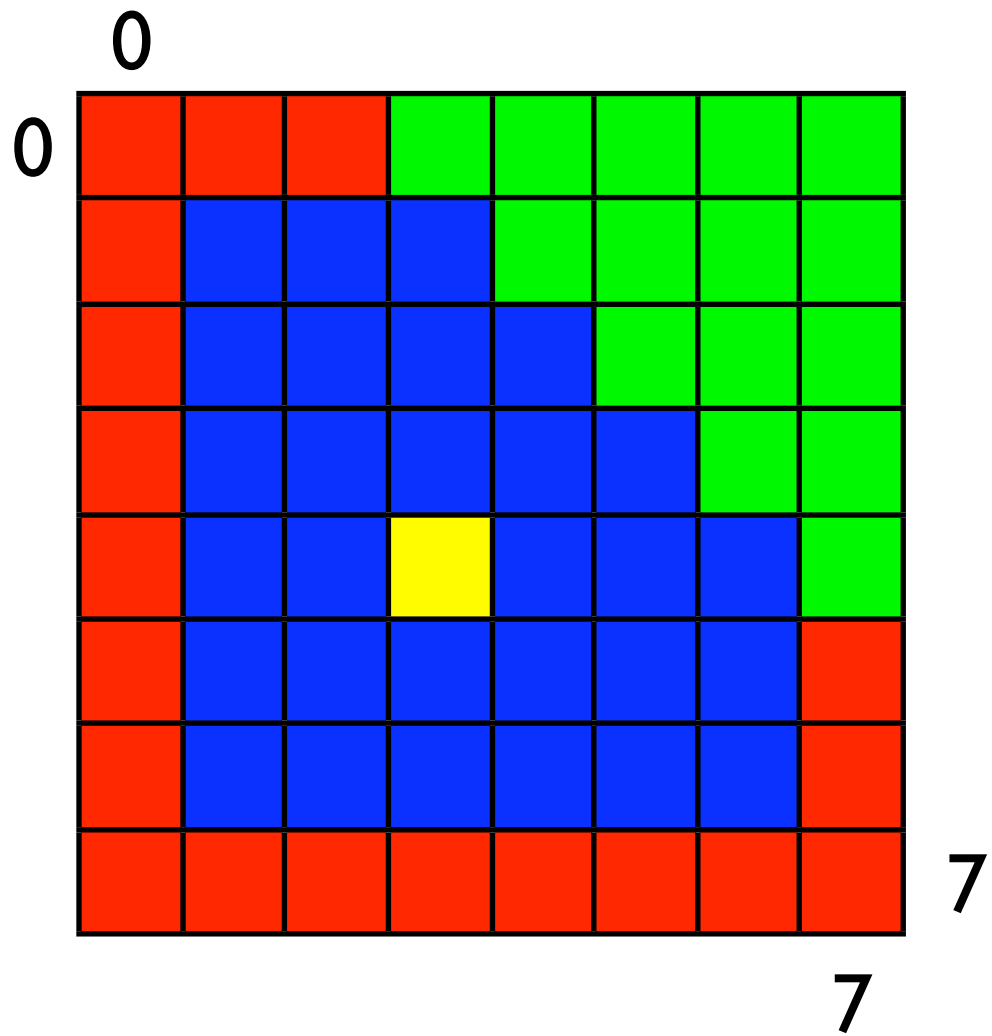
Befüllen von Regionen mit gleichem Farbwert mit neuer Farbe



„Region Growing“

fill(3,4,yellow)

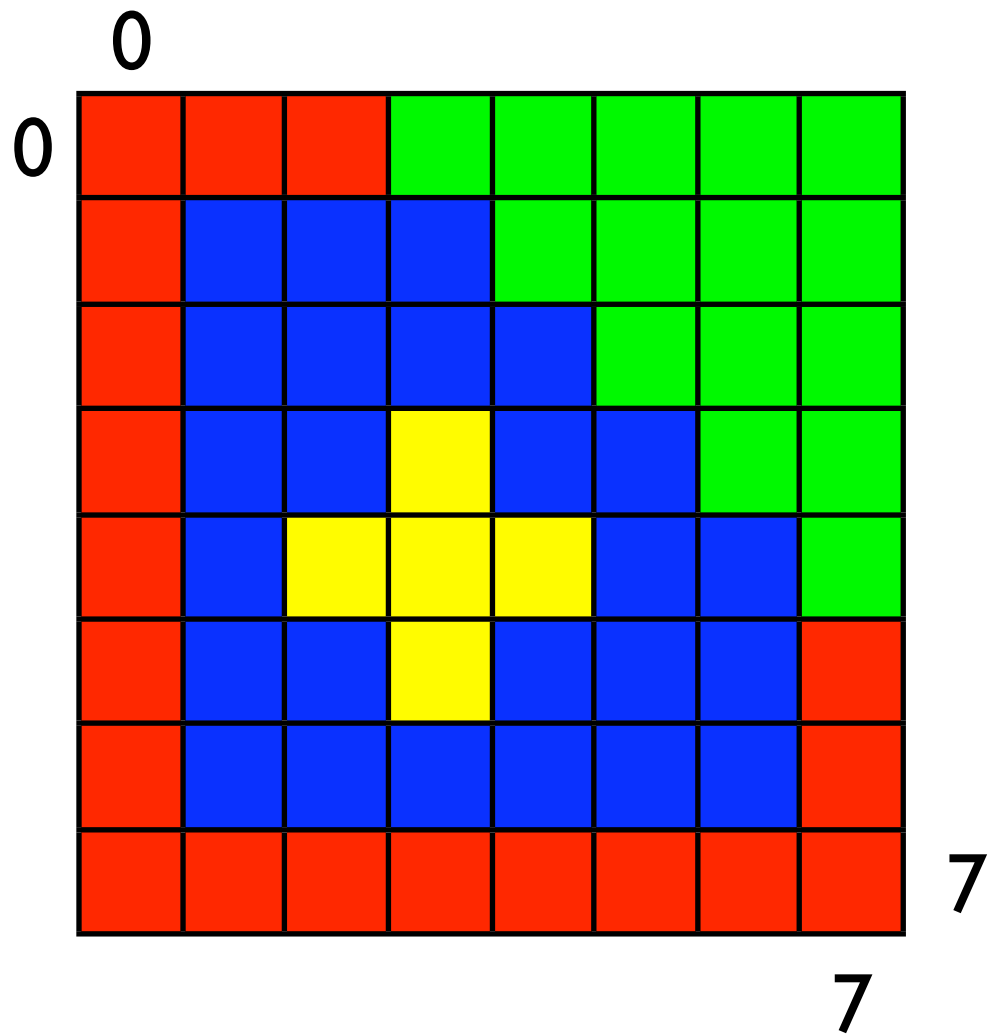
Ausgehend vom Punkt (3,4) werden benachbarte Pixel mit gleichem Farbwert gelb gefärbt



„Region Growing“

fill(3,4,yellow)

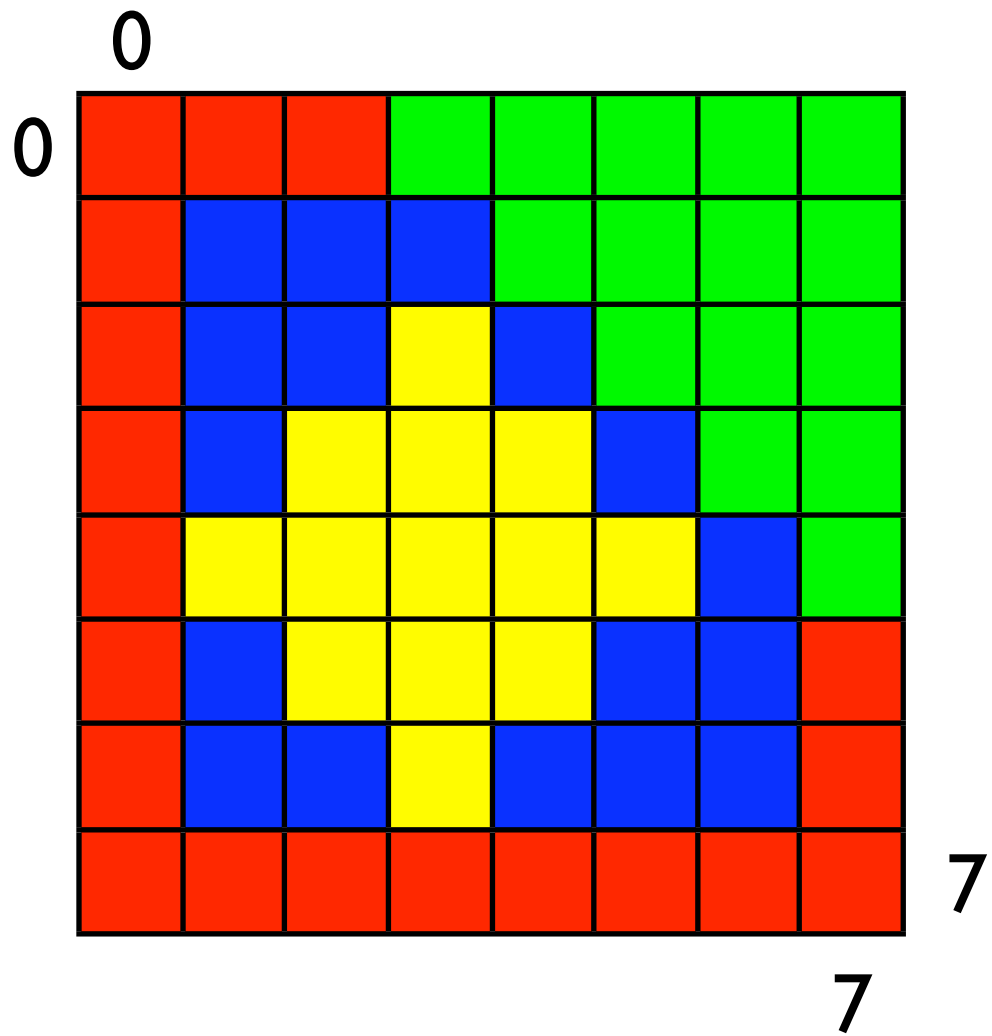
Ausgehend vom Punkt (3,4) werden benachbarte Pixel mit gleichem Farbwert gelb gefärbt



„Region Growing“

fill(3,4,yellow)

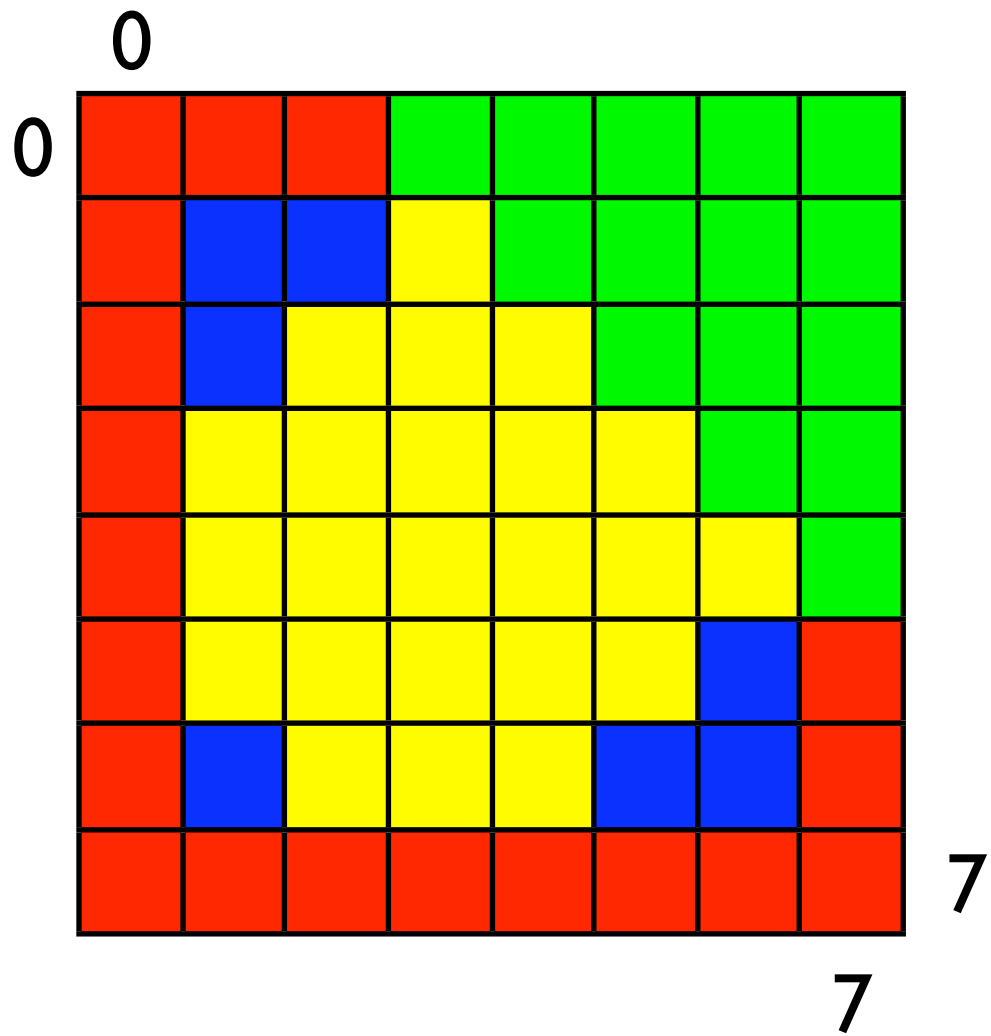
Ausgehend vom Punkt (3,4) werden benachbarte Pixel mit gleichem Farbwert gelb gefärbt



„Region Growing“

fill(3,4,yellow)

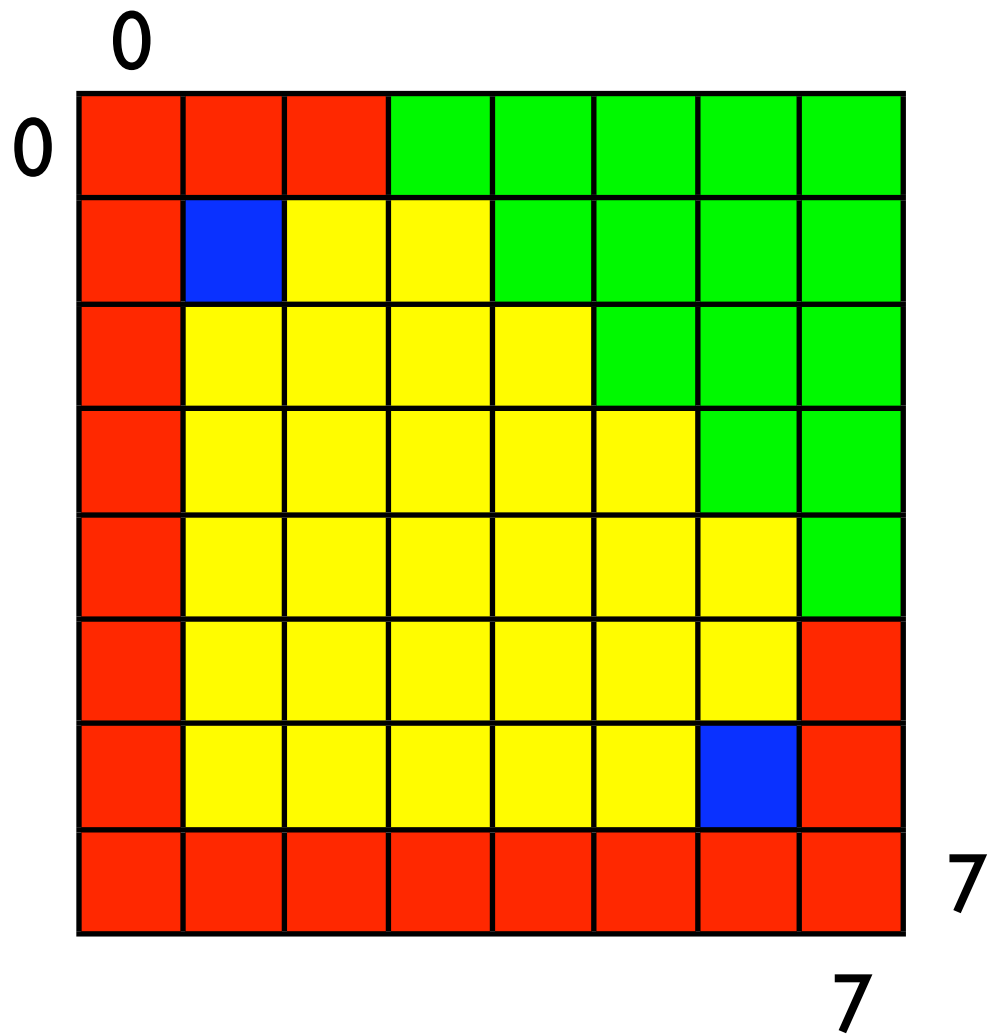
Ausgehend vom Punkt (3,4) werden benachbarte Pixel mit gleichem Farbwert gelb gefärbt



„Region Growing“

fill(3,4,yellow)

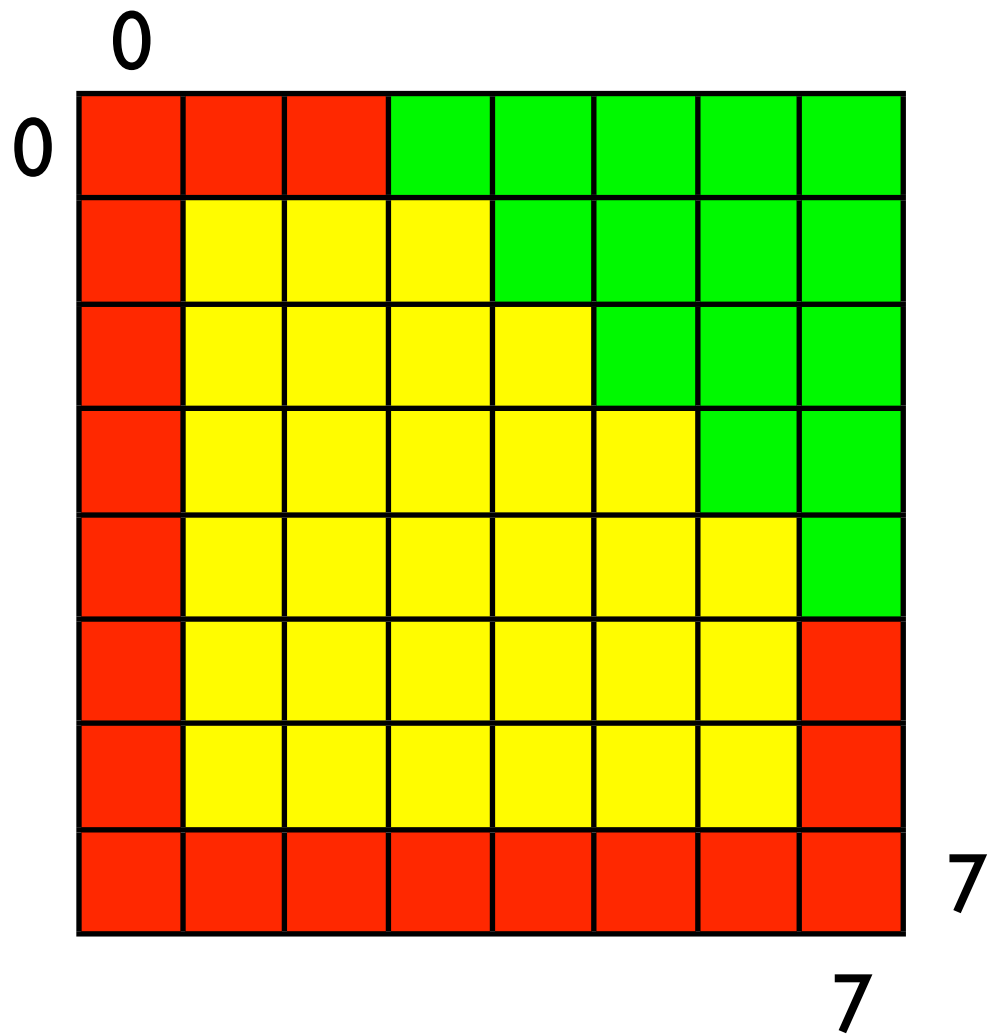
Ausgehend vom Punkt (3,4) werden benachbarte Pixel mit gleichem Farbwert gelb gefärbt



„Region Growing“

fill(3,4,yellow)

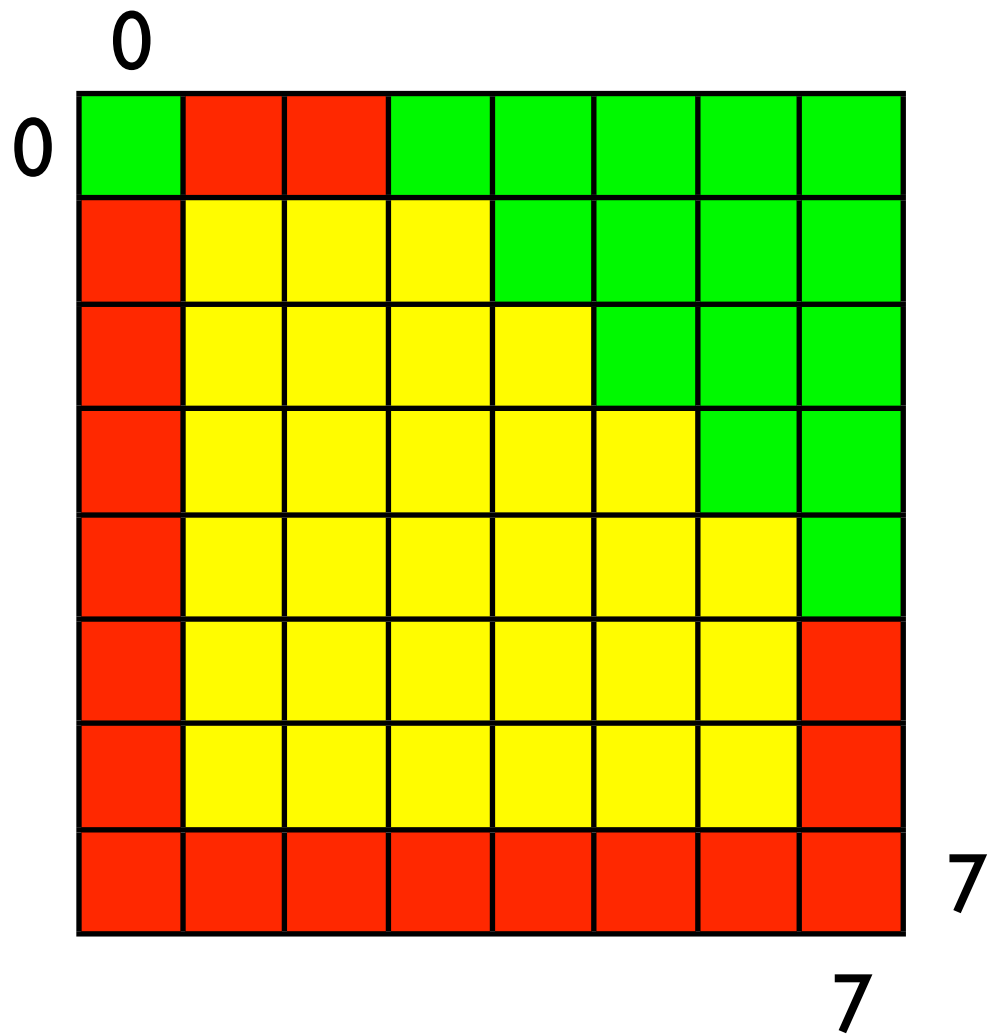
Ausgehend vom Punkt (3,4) werden benachbarte Pixel mit gleichem Farbwert gelb gefärbt



„Region Growing“

fill(3,4,yellow)

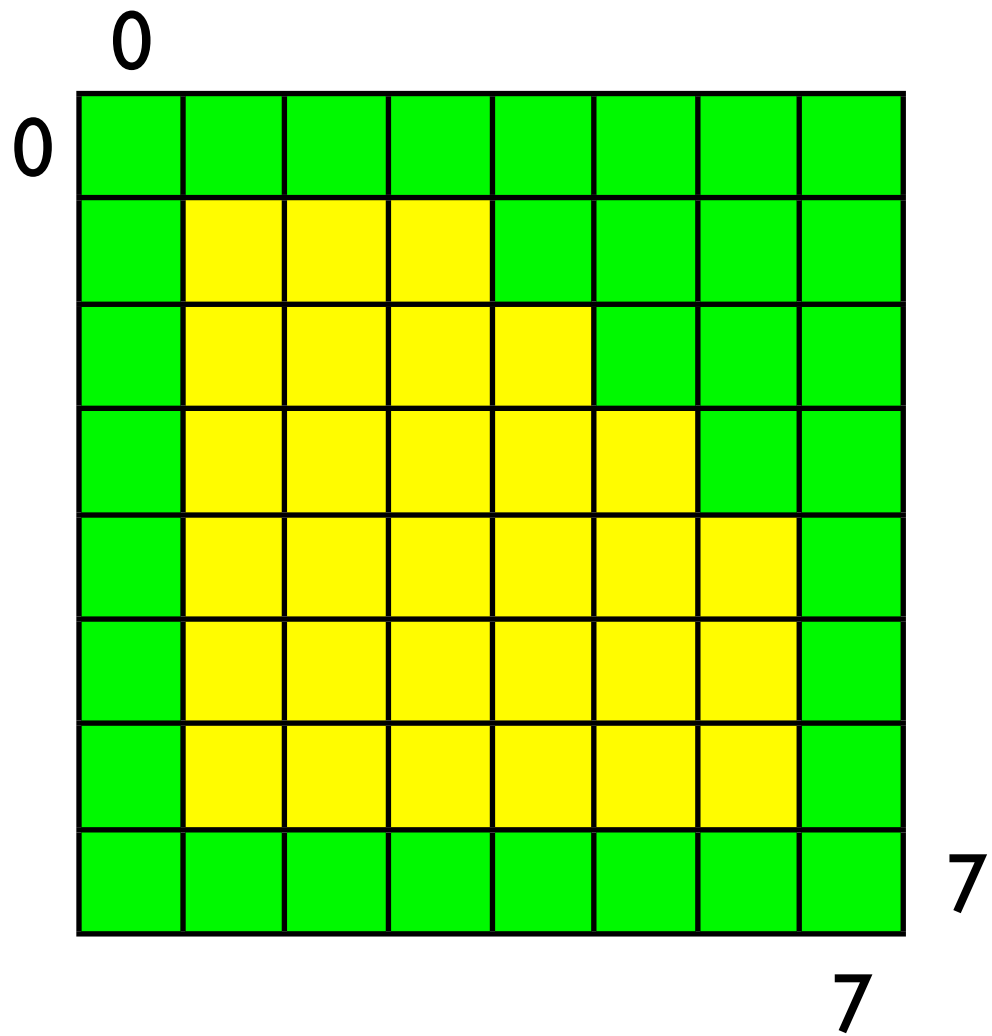
fill(0,0,green)



„Region Growing“

fill(4,5,yellow)

fill(0,0,green)

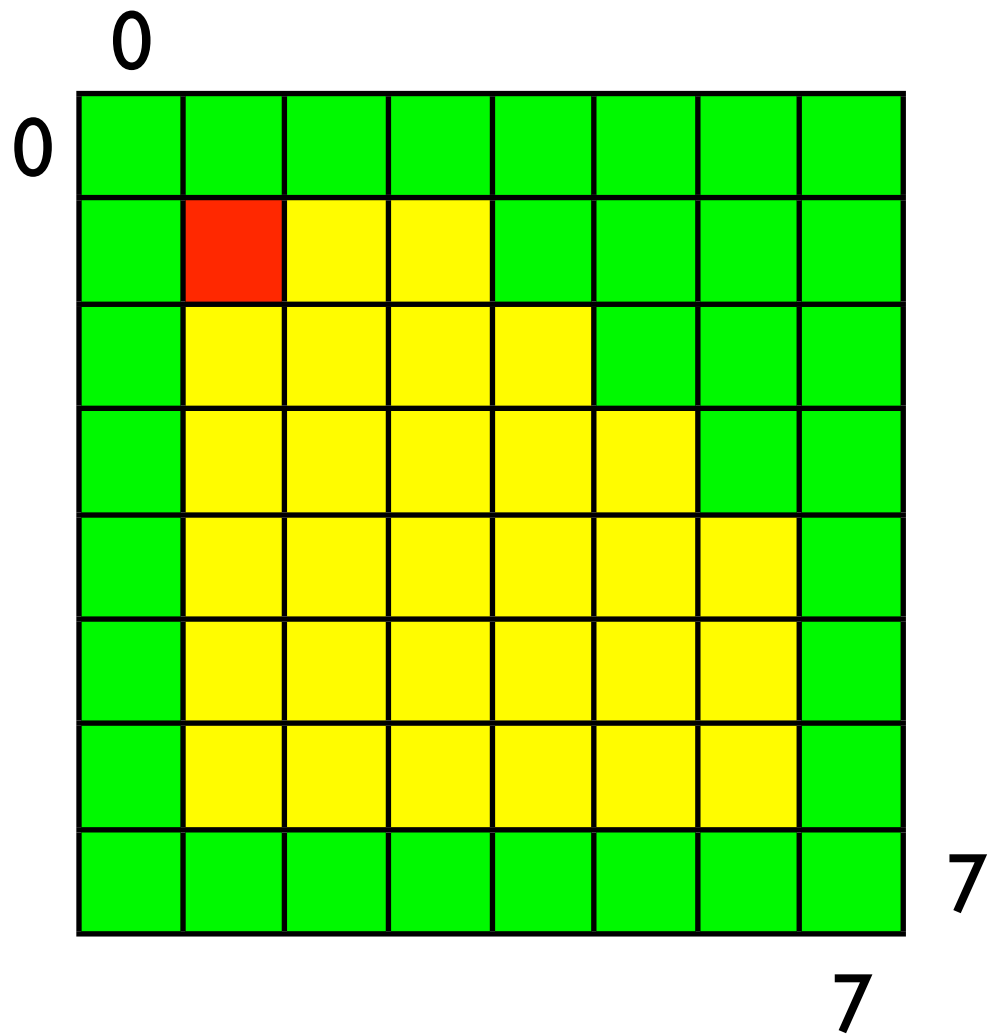


„Region Growing“

fill(3,4,yellow)

fill(0,0,green)

fill(1,1,red)

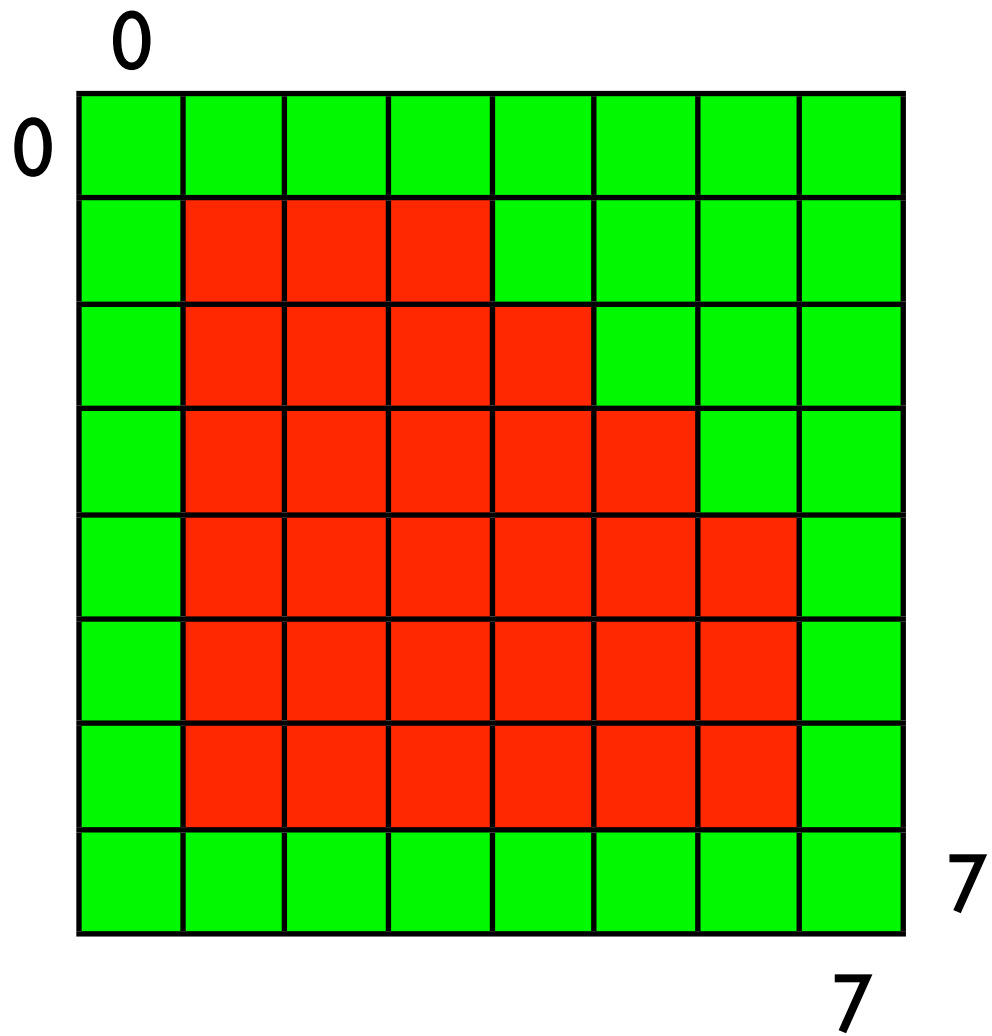


„Region Growing“

fill(3,4,yellow)

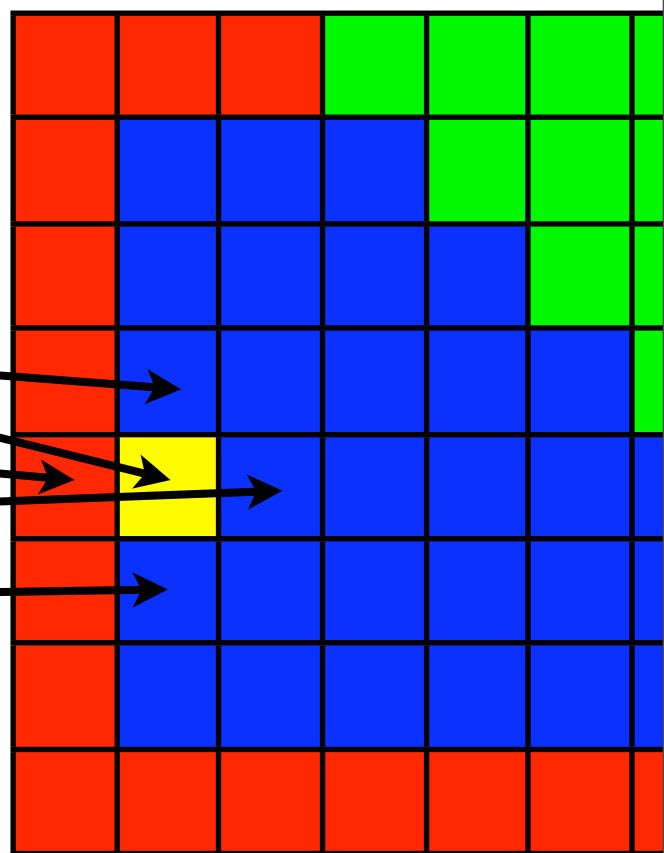
fill(0,0,green)

fill(1,1,red)



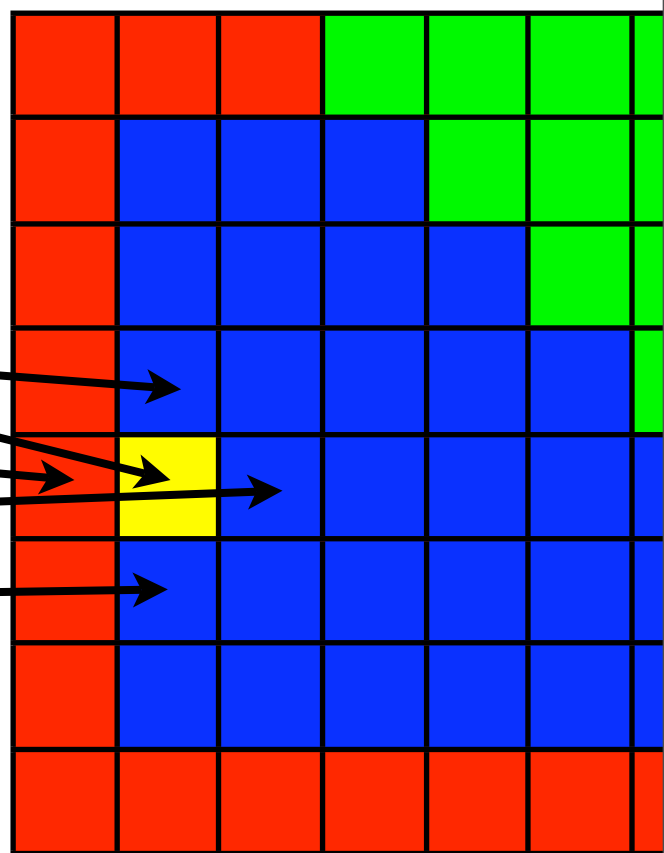
Rekursiver Aufruf

```
internal[i][j] = put;

return fill(i-1,j,put,replace)
 + fill(i,j-1,put,replace)
 + fill(i,j+1,put,replace)
 + fill(i+1,j,put,replace)
  + 1;
```



Rekursiver Aufruf

```
internal[i][j] = put;

return fill(i-1,j,put,replace)
 + fill(i,j-1,put,replace)
 + fill(i,j+1,put,replace)
 + fill(i+1,j,put,replace)
  + 1;
```



Rekursiver Aufruf

```
if (internal[i][j] == replace )
{
   internal[i][j] = put;

   return fill(i-1,j,put,replace)
    + fill(i,j-1,put,replace)
    + fill(i,j+1,put,replace)
    + fill(i+1,j,put,replace)
     + 1;
} else return 0;
```